(12) **UK Patent Application** (19) **GB** (11) **2 366 957** (13) **A**

(43) Date of A Publication 20.03.2002

(21) Application No 0112765.3

(22) Date of Filing 25.05.2001

(30) Priority Data
(31) 0012782   (32) 26.05.2000   (33) GB
(31) 0022674   (32) 15.09.2000

(71) Applicant(s)
Roke Manor Research Limited
(Incorporated in the United Kingdom)
Old Salisbury Lane, ROMSEY, Hampshire, SO51 0ZN,
United Kingdom

(72) Inventor(s)
Mehul Mehta

(74) Agent and/or Address for Service
Siemens Shared Services Limited
Siemens House, Oldbury, BRACKNELL, Berkshire,
RG12 8FZ, United Kingdom

(51) INT CL^7
H04Q 7/32 // G06F 15/78

(52) UK CL (Edition T)
H4L LECCP
G4A AFL

(56) Documents Cited
WO 01/02954 A1   US 5999990 A

(58) Field of Search
UK CL (Edition T) G4A AFL , H4L LECCP LECY LEP
LERX LETXX
INT CL^7 G06F 9/44 9/445 15/78 , H04B 1/40 , H04Q
7/32
ONLINE DATABASES: WPI, EPODOC, JAPIO, INSPEC

(54) Abstract Title
Reconfiguration manager

(57) A device such as a software defined radio (SDR)
equipment (e.g. a mobile telephone) is reconfigured
using a reconfiguration manager 20 which controls the
procedures and mechanisms for downloading and
implementing software modules from a library of stored
modules. The baseband or RF subsystems may be
reconfigured for example for operation in different
networks. Reconfiguration is achieved without affecting
current operation by maintaining an active chain of
software blocks, preparing in parallel a shadow chain of
software blocks and then switching to the shadow chain
which then becomes the active chain. The
reconfiguration manager comprises three units: a
reconfiguration handler 62 for receiving reconfiguration
instructions from a higher authority, a run-time control
signal handler 72 responsible for implementing
reconfiguration, and a run-time monitoring and status
handler 74 for implementing status requests and for
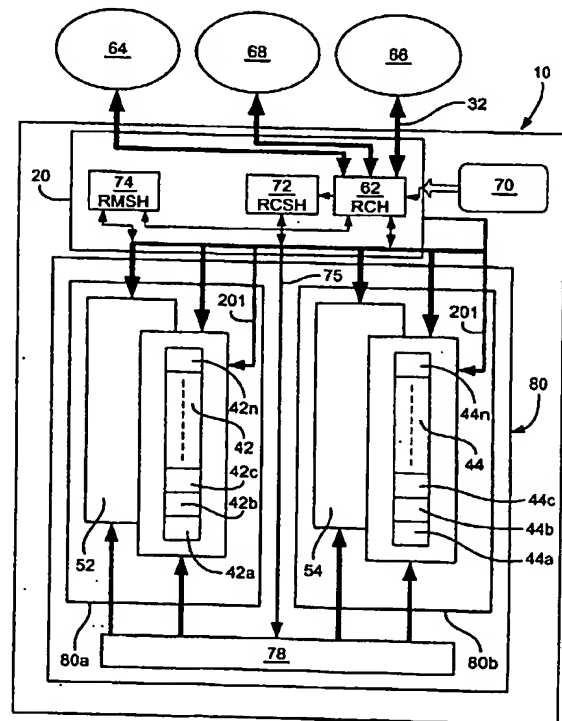reading acknowledgements and status information.

Fig.2

GB 2 366 957 A

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

This print takes account of replacement documents submitted after the date of filing to enable the application to comply
with the formal requirements of the Patents Rules 1995

Fig.1

40

64

33

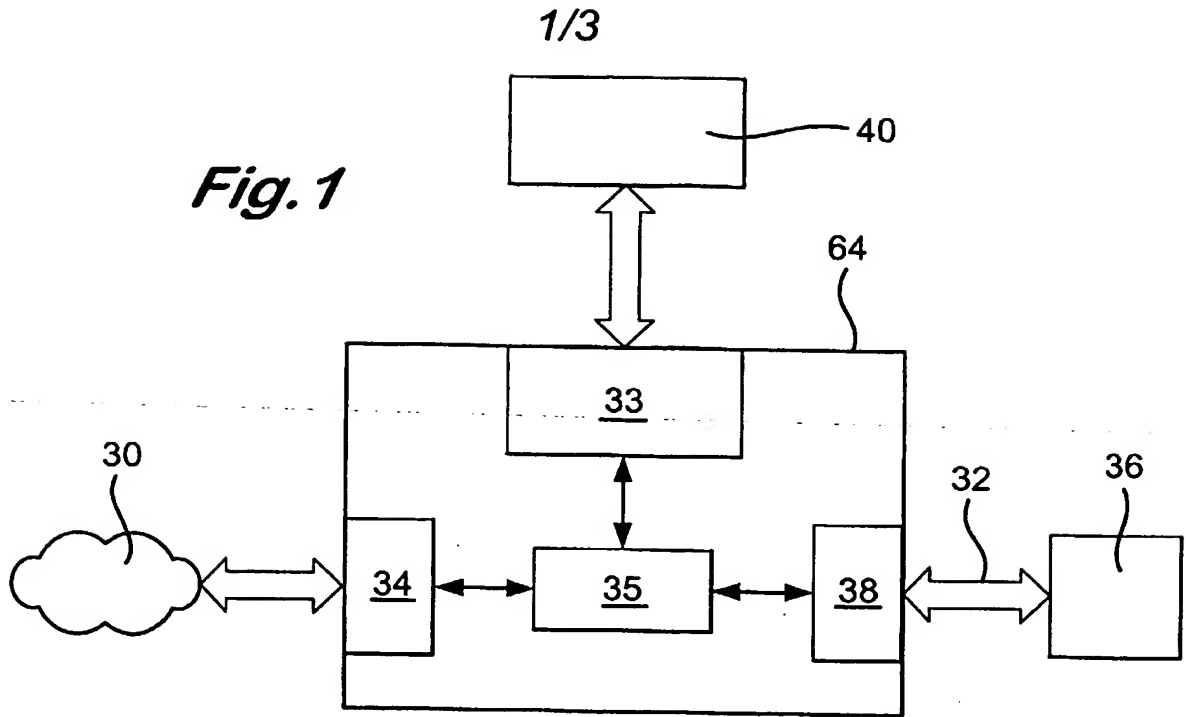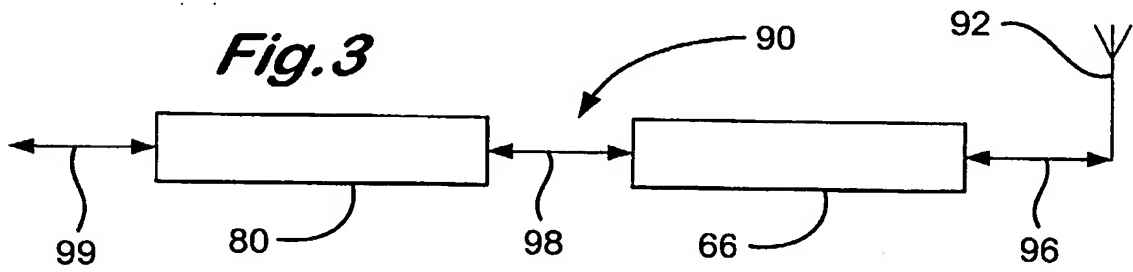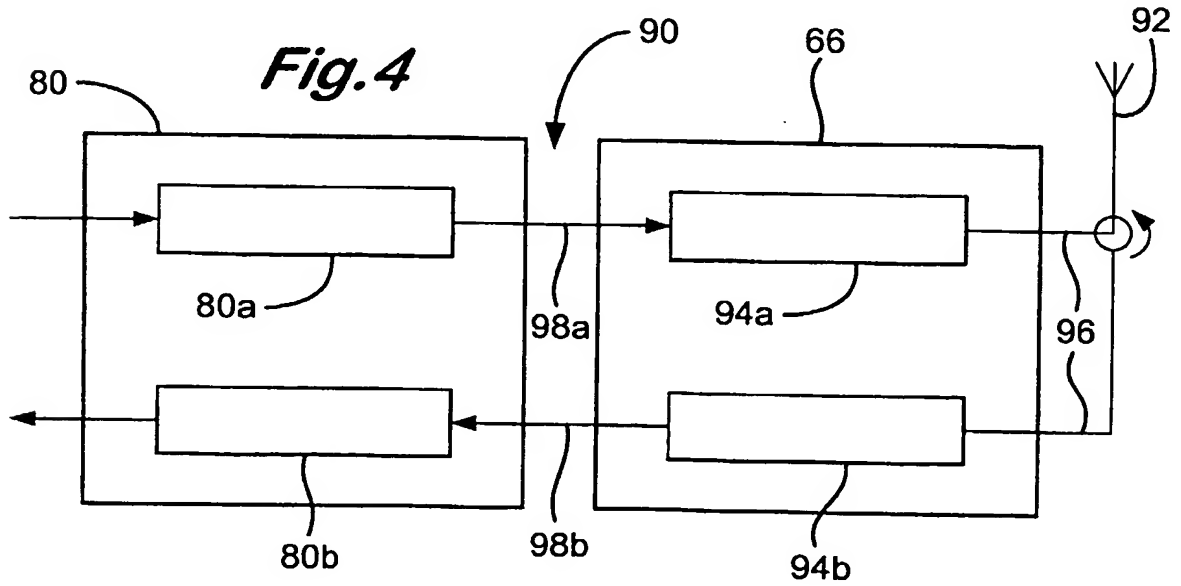30

34   35   38

32   36

Fig.3

90   92

80   98   66   96

99

Fig.4

80   90   66   92

80a   98a   94a   96

80b   98b   94b

Fig.2

Fig.5

```
→ [101a] → [101b] → [101c] - - → [101n] →   80a
```

```
← [102a] ← [102b] ← [102c] - - [102n] ←   80b
```

Fig.6

BPC_A

C(A) 201 140

124 122 134

128 112 135

I(A) P O(A)

126 110 136

129 120 130 138

132

Fig.7

304 → 305 → 306 → 307

302

303 308

301
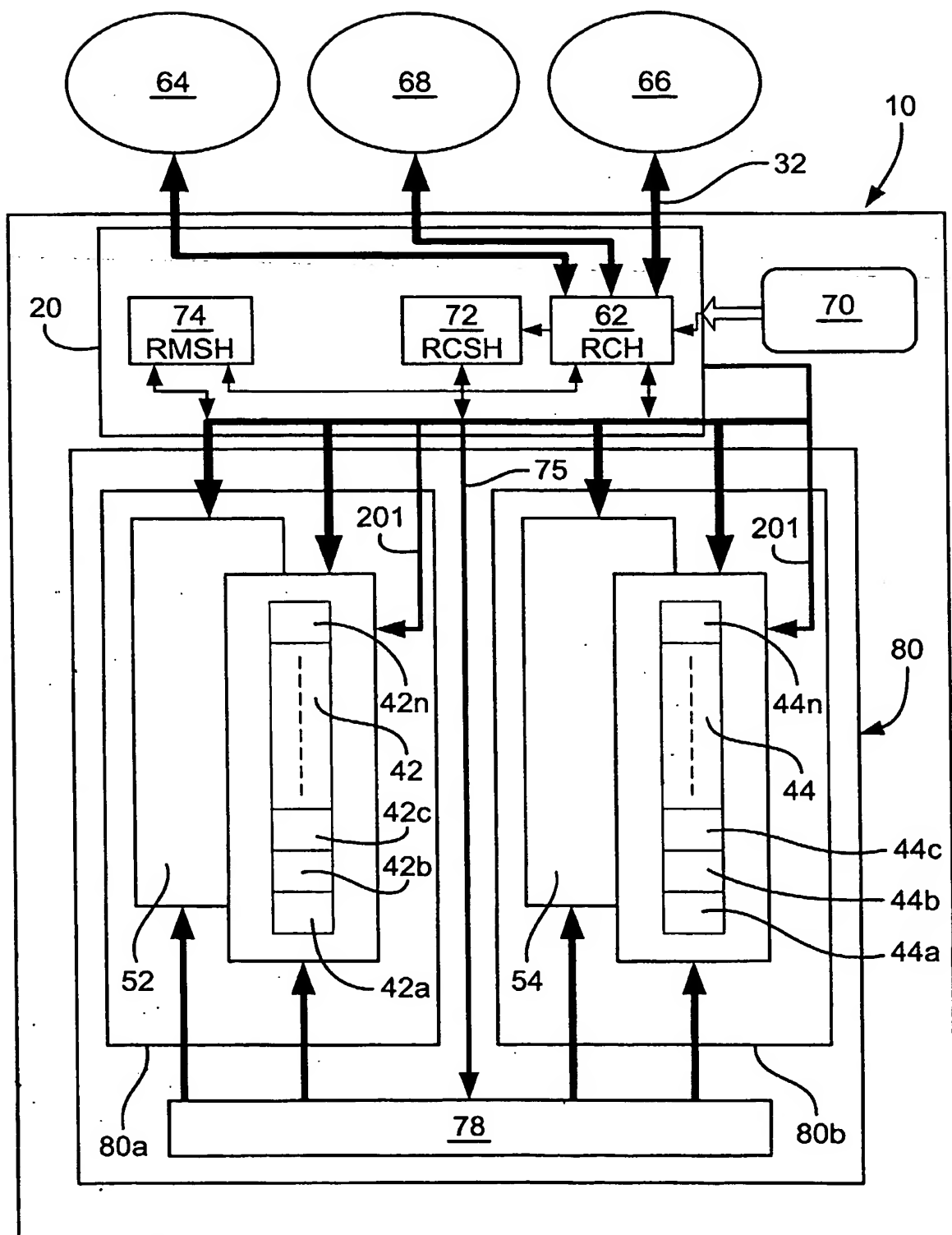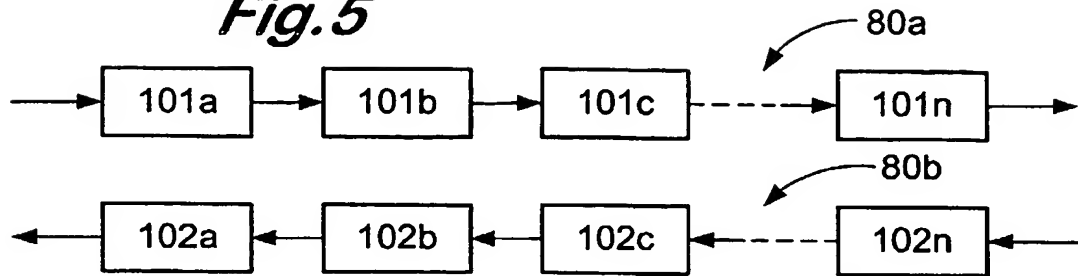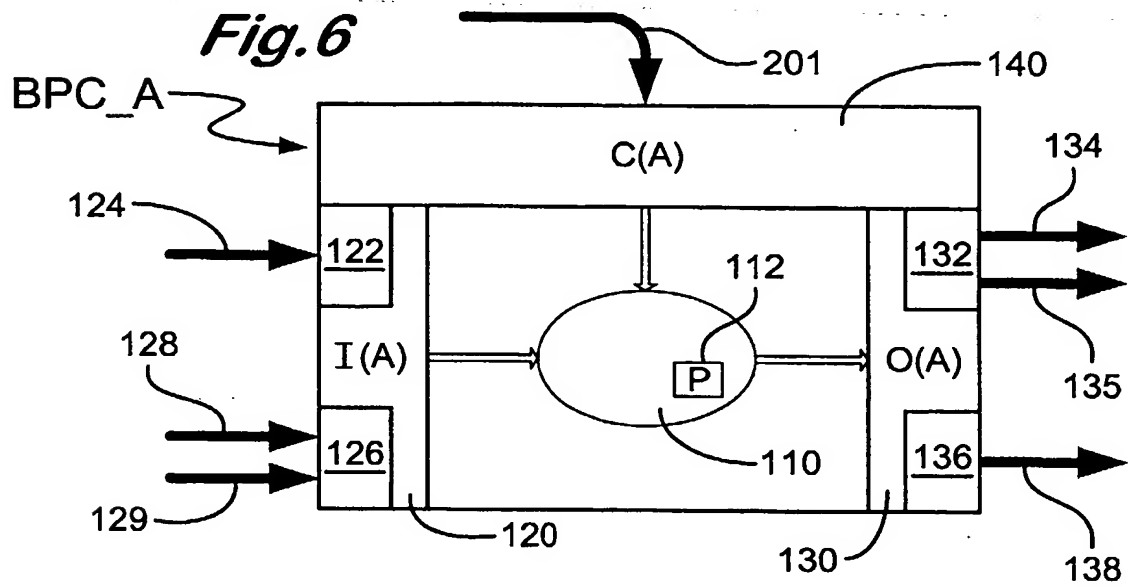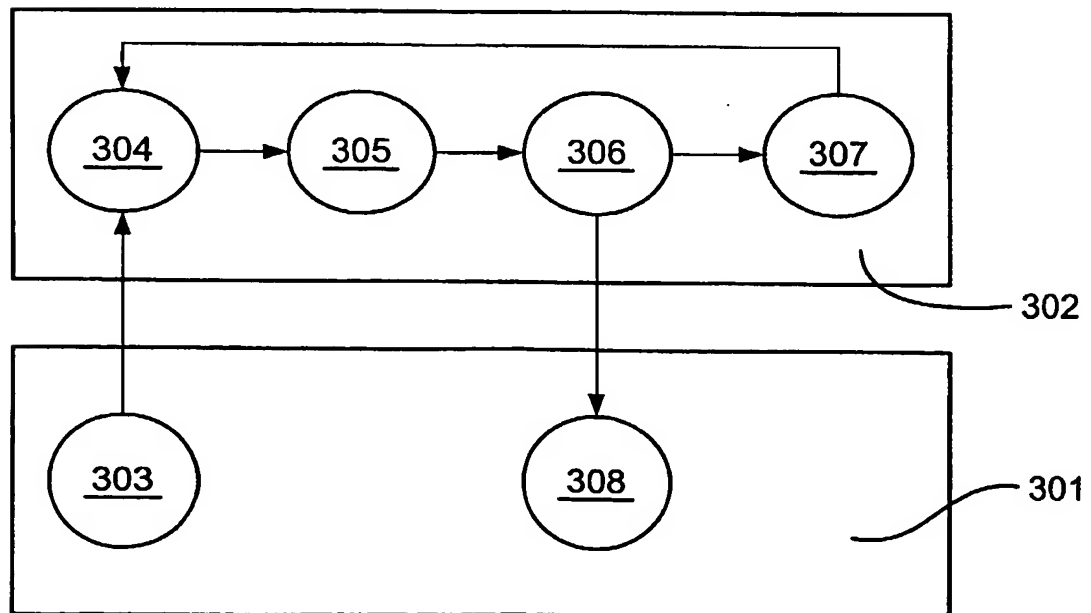
RECONFIGURATION MANAGER

The present invention relates to a Reconfiguration Manager (RM) for a reconfigurable system, such as a software defined radio (SDR) equipment. An SDR equipment is any piece of equipment incorporating an SDR radio. In one common

5 application, the SDR equipment is a portable SDR user terminal, incorporating a mobile telephony function. In particular, the present invention relates to, and will be discussed with reference to, a reconfiguration manager for a baseband subsystem of a software-defined radio terminal.

The following sources may be relevant to certain parts of the present description,

10 particularly at locations indicated by their reference numeral. The whole contents of each of the following sources is incorporated herein by reference. [1] http://www.ist-trust.org. [2] N. Drew et al, Reconfigurable Mobile Communications: Compelling Needs and Technologies to support Reconfigurable Terminals, IST Mobile Communications Summit 2000, Galway, Ireland, October 2000. [3] T. Farnham et al, System Aspects of Software

15 Defined Radio within the TRUST project, IST Mobile Communications Summit 2000, Galway, Ireland, October 2000. [4] M. Siebert, "Design of a Generic Protocol Stack for an Adaptive Terminal," Karlsruhe Workshop on Software Radios, March 2000. [5] K. Moessner, S. Vahid, R. Tafazolli, "A Minimum Air-Interface Implementation for Software Radio based on Distributed Object Technology, " IEEE ICPWC 1999.

20 A software defined radio (SDR) terminal is typically a portable device which can be programmed to perform any of a variety of functions. Some of these may be conventional "communications" functions such as mobile telephone, or internet access whereas others may be less obviously communications related, such as a portable games device/console. This latter may, for example, be configurable by loading games software

25 or updates over the air.

SDR is based on flexible software architecture. This allows radio equipment (e.g. user terminal, base stations etc.) to be reconfigured by downloading and implementing appropriate software. In other words, the radio equipment is software re-definable. As a result, a SDR terminal can operate several different wireless communications standards,

30 and also modify its behaviour whilst remaining in a given standard.

Existing software defined radio terminals are configured according to the requirements of the network, and of the terminal itself. For example, The network may

configure the radio according to the capacity metrics of the network, or the air interface characteristics, available bandwidth or available radio spectrum. Typical air interfaces that could be used in such applications include: WAN such as GSM, DECT, UMTS; IEEE802.11a/b; and HIPERLAN/2. The terminal may itself impose certain restrictions on

5 the available configurations, for example, according to its own capabilities, such as its power profile, processing power and memory capacity. Software defined radio terminals may take one of many different formats, for example, they may be hand-held, palm-top, lap-top or desk-top. Each of these will have its own characteristics which will influence the configuration applied to it.

10 Software defined radio terminals are often set up to include mobile telephony function. Either the mobile terminals or the base station, or both of them, may be a suitably configured software defined radio. However, software defined radio terminals are not limited to such application and may, indeed, be configured as any communicating device.

15

Current research is focusing on the needs and design of a reconfigurable terminal in the context of a composite radio access environment [1] [2], particularly on commercial wireless communications and reconfiguration of the terminal from one standard to another, e.g., from GSM to HiperLan/2. The present invention may be applicable in the

20 context of the baseband sub-system for the CEC 5th Framework IST TRUST (Transparent Reconfigurable Ubiquitous Terminal) project.

Co-pending UK patent application No. GB 01.... (Agent's reference: 2000P04892GB), or co-pending International patent application No. PCT/GB01/....

25 (Agent's reference: 2000P04892WO), each filed of even date herewith and incorporated herein by reference, discloses a management module which may be provided within a software defined radio terminal, to operate the configuration of the terminal according to user, network and terminal requests and constraints. As shown in Fig. 1, the management module (MM) 64 is effectively an intelligent management entity of the SDR equipment.

30 The management module 64 communicates with an associated network 30 through the network interface 34, with the SDR equipment 36 through the equipment interface 38 and with the operator 40 through the operator interface 33. It communicates in accordance

with its command and control algorithms 35. The associated network may be internal to the equipment of which the MM 64 forms part, or may be an external network, of which the equipment forms part.

Management module MM 64 is linked to the associated SDR equipment 36 by an
5 equipment interface 38 carrying signals 32. Both the management module and the SDR equipment may be respectively implemented in either hardware or software. Operator interface 33 defines formal procedures for signalling and control to/from the operator 40, which may be man or machine.

Communication with the operator 40 may include operator interface control, re-
10 configuration requests and acknowledgements, operator profiles, application and quality of service control, and power management such as user interrupt and signalling functions.

Communication 32 with the SDR equipment 36 may include power management, re-configuration negotiations and administration, terminal mode control, software download control, terminal application control, run-time control signalling, terminal class
15 (capability list) and terminal resources (RF, input/output, hardware).

Communication with the network 30 may include higher layer signalling, network negotiations, power management, re-configuration management and administration, run-time control signalling, mode control and software download.

All of the entities operator 40, network 30 and terminal 36 may request
20 reconfiguration.

The management module 64 defines the functionality, behaviour and state of the whole reconfigurable system, comprising the management module itself, the associated equipment 36, the network 30 and the operator 40. It administers reconfiguration, including all the associated network and radio signalling and control mechanisms. In
25 addition, it provides formal procedures for handling and implementing command and control signals from the operator (equipment user), the network, and the equipment itself. In the above referenced UK and International patent applications, there is disclosed a management entity which allows the user to reconfigure the terminal as he desires, as and when required.

30 The present invention provides a novel software architecture of a baseband sub-system for a Software Defined Radio equipment. The architecture is flexible, such that it

4

can reconfigure itself in response to operator or system (network or equipment) reconfiguration requests, upon the availability of appropriate software.

Certain embodiments use an object-oriented approach, whereby the functionality of a given baseband module can be changed by instantiating appropriate classes. The
5 formulation of baseband classes and their associated attributes will be discussed in relation to certain embodiments of the present invention. In the described embodiments, a higher authority, management entity 64, is assumed, to regulate, administer and manage baseband reconfiguration. This entity, together with the signalling required to perform reconfiguration, is briefly described above, and an example of such entity is further
10 discussed in the UK and International patent applications referenced above.

Herein, "layers" refers to the conceptual layers of a communications protocol, for example the seven-layer OSI reference model, of which the lowest layer is the physical layer, dealing with the transmission of bits to and through a physical medium, while the highest layer is the application layer, which comprises the overall application as seen by
15 the user and miscellaneous protocols for common activities.

The software architecture of the present invention is not restricted to the baseband sub-system. The concept of changing the functionality, behaviour, and interfaces of a certain processing unit, by instantiation of appropriate classes within an Object-Oriented (OO) design, may also be equally applicable to higher layers (i.e., above the physical
20 layer) [4], [5], and to other subsystems within the physical layer, such as the RF subsystem.

From the baseband perspective, there are two types of reconfiguration. Both of these reconfiguration scenarios are possible under the system of the present invention, and both will be described further with reference to the present invention.

25 Total reconfiguration is reconfiguration from one standard to another, i.e., inter-standard reconfiguration. For example, from GSM to UMTS. This reconfiguration corresponds to exhaustive changes in the functionality, behaviour, and interfaces of the constituent modules.

Partial reconfiguration is a reconfiguration without changing the operating
30 standard, i.e., intra-standard reconfiguration. For example, if the reconfigurable system is defined in terms of configurable software modules, a partial reconfiguration may be one in

which one or more modules are reconfigured, for example, in order to improve quality of service whilst remaining on the current operating standard.

According to an aspect of the invention, there is provided an adaptive baseband
5 subsystem which is reconfigurable, for example, using a reconfigurable baseband software as described below.

According to an embodiment of the invention, a reconfiguration manager (RM) is provided. It defines formal procedures and control mechanisms for reconfiguring the baseband sub-system of a reconfigurable system such as a SDR terminal. The RM
10 provides an overall physical layer interface with the higher layers which may themselves be controlled by a management entity such as the Management Module (MM) discussed above.

The reconfiguration manager may implement run-time signalling and reconfiguration protocols at baseband.

15 The baseband sub-system of a SDR terminal is inherently flexible, such that the functionality, behaviour and the interfaces of the constituent modules can be re-programmed by software. The baseband is re-definable by software, and is therefore also referred to as the Reconfigurable Baseband (R-BB). The potential capability to reconfigure the baseband from one standard to another (e.g., GSM to UMTS), and also, to
20 change its behaviour within a given standard is the objective of a reconfigurable baseband sub-system.

The present invention therefore provides methods and apparatus as defined in the appended claims.

25

The above, and further, objects, characteristics and advantages of the present invention will be further described with reference to certain embodiments, given by way of examples only, in conjunction with the accompanying drawings, in which:

Fig. 1 represents a management module for a software defined radio equipment, as
30 discussed above;

Fig. 2 shows a schematic diagram of a reconfigurable baseband (R-BB) subsystem according to an embodiment of the present invention;

Figs. 3-4 schematically show physical layers of general radio equipment;

Fig. 5 illustrates chains of software blocks used in the physical layer of a radio equipment according to an embodiment of the present invention;

Fig. 6 schematically illustrates the structure of a baseband processing cell 5 according to an aspect of the present invention; and

Fig. 7 shows a state transition diagram for partial and total reconfiguration according to embodiments of the present invention.

In order to manage and administer baseband reconfiguration, a management entity 10 is required for the R-BB to implement protocols for reconfiguration and regulate appropriate signalling to/from the baseband and also within it. This role may be fulfilled by the reconfiguration manager (RM) of the present invention.

The RM according to the present invention has the following features.

It is the overall authority of the physical layer.

15 It provides a formal interface for the physical layer with the upper layers.

It controls the baseband and RF sub-systems.

It negotiates reconfiguration requests with the SDR terminal's equipment authority (EA), which may be a management module (MM) as discussed above.

It carries out a RF capability list as part of the reconfiguration request negotiations.

20 It calculates the complexity of a given reconfiguration request.

It administers baseband reconfiguration following successful completion of negotiations with the EA.

It interacts with the Software Download Module (SDM) of the terminal for downloading new software.

25 It maintains a local library of the baseband software. The library includes the baseband configuration map. This is the overall baseband module identity, interface, and inter-connection definition list for the standard operating on the terminal.

It validates the reconfigured baseband transceiver chain before switching it on.

It implements incoming higher layer signalling to the baseband.

30 It monitors the run-time performance and status of baseband modules.

Fig. 2 shows a schematic diagram of a reconfigurable baseband (R-BB) subsystem 10 according to an embodiment of the present invention. Reconfiguration Manager (RM) 20 is the overall authority of the R-BB sub-system. It is responsible for negotiating reconfiguration, creating active and shadow transceiver chains, and controlling the run-

5 time behaviour of each module. The RM 20 also controls the RF sub-system 66 through appropriate signalling 32, making it the overall physical layer authority.

Figure 2 schematically shows a RM 20 consisting of three units, namely, a reconfiguration handler (RCH) 62; a run-time control signalling handler (RCSH) 72 and a run-time monitoring and status handler (RMSH) 74.

10 The re-configuration handler RCH 62 deals with baseband reconfiguration. Any signalling and negotiations related to reconfiguring any part of the baseband will be handled by this entity. It is in communication with the equipment authority (EA) 64, the RF subsystem 66, the software download module 68, the software library 70, the RMSH 74, the RCSH 72, and a receive-transmit baseband module chain 80.

15 The run-time control signalling handler RCSH 72 is responsible for implementing higher layer signalling and control signalling at the baseband. This unit deals with run-time signalling, which will be standard specific. However, it will also receive signals from the RCH 62 in order to regulate the behaviour of the baseband while it is being reconfigured. This will lead to modifications in the run-time control signalling. The

20 RCSH 72 is in communication with the equipment authority 64, the baseband module chain 80 and receives signals from the RCH 62.

The run-time monitoring and status handler RMSH 74 implements run-time status requests on the baseband subsystem 80 and reads in the respective acknowledgements. In the illustrated embodiment, the baseband subsystem is made up of constituent baseband

25 modules. The RMSH therefore has a supervisory role to oversee the run-time behaviour of each baseband module. In addition, it also collates auxiliary outputs from the baseband subsystem and reports it to the equipment authority EA 64. Accordingly, it is in communication with the equipment authority 64, the RCH 62 and the baseband subsystem 80.

30 The reconfiguration manager 20 is the overall authority of the physical layer, that is, the part of the system that is directly involved in the transfer of data, for example in the form of binary digits. In the example system described above, the physical layer may

comprise the baseband subsystem 80, itself comprising module chains 42, 44, 52, 54, and an RF subsystem 66.

The equipment authority (EA) 64 has overall authority for the whole SDR equipment, and so has an over-riding authority over the reconfiguration manager 20. The 5 equipment authority (EA) referred to in this description may correspond to the management module (MM) discussed elsewhere.

To perform a reconfiguration of the physical layer, that physical layer will have to be defined in functionality descriptions. The actual implementation of the functions described herein is of minor importance, and a suitable implementation of any of the 10 described functions will be apparent to one skilled in the art. However, the presently described aspect of the present invention describes novel and inventive combinations of functionality and interfaces, behaviour and features.

As shown in Fig. 3, any radio equipment, such as an SDR terminal, which transmits and/or receives digital data over RF signals, will include a physical layer 90 15 comprising an antenna 92, an RF subsystem 66 and a baseband subsystem BB 80. Radio signals 96 received by the antenna 92 are converted by the RF sub-system 66 into a baseband signal 98, and the baseband subsystem 80 extracts the required data 99 for supply to the remainder of the equipment. Similarly, in the opposite sense, data 99 for transmission is sent to the baseband subsystem 80 which generates a corresponding 20 baseband signal 98, which is itself then forwarded to the RF subsystem 66 for encoding onto a radio signal 96 for transmission from the antenna 92.

Fig. 4 illustrates a typical physical layer sub-system 90 for a SDR terminal, in greater detail. As can be seen in the figure, the receive and transmit paths are substantially separate, and may combine at the antenna. The transmit 94a and receive 94b RF 25 subsystems are each tuneable in software. They receive and supply, respectively, baseband signals 98a, 98b from/to transmit, and receive, baseband subsystems 80a, 80b which may be embodied within a DSP (digital signal processor).

As shown in Fig. 5, according to another aspect of the present invention and as illustrated also in Fig. 2, each of the transmit and receive baseband subsystems 80a, 80b 30 may be composed of a number of modules 101a-101n; 102a-102n in series, each of which may be loaded with a selected software block. The subsystems 80a and 80b may then be referred to as receive and transmit baseband module chains. Each of the modules may be

loaded with a downloaded software block, received over the air interface by the SDR terminal equipment, or previously stored in software library 70 of the SDR terminal equipment.

The software download module (SDM) 68 is provided to control the downloaded
5 software blocks.

An example of the reconfiguration that may be applied to the baseband system is the conversion from the GSM standard, which operates at 900MHz with a 2kHz channel, to the UMTS standard which operates at 2.5GHz with a 5MHz channel. In the case of such reconfiguration, all of the processing algorithms will need to be changed, meaning
10 that the SDR terminal, and particularly the RF and baseband subsystems 66, 80, would need to be reconfigured.

RCSH 72 performs all the required signalling, including control signalling to and from the base station, timing slots etc. The RCSH 72 implements the control signalling required by the agreed air interface, that is, the air interface agreed for use by the user and
15 the network. The RCSH 72 handles all signalling, but is not involved in data transmission. The coarse features of the data processing are regulated by this signalling.

The RMSH 74 performs the internal monitoring of all baseband 80 and RF subsystem 66 operations. It checks on each software block 101, 102 of the baseband chains, monitors features such as operation and power consumption, and reports its
20 findings to the EA 64, which may then decide to send a request for reconfiguration to the reconfiguration handler.

Whilst the flexible architecture of SDR terminals is typically restricted in its scope to the baseband sub-system, it is conceptually suitable for reconfigurable higher layers as well, assuming a conventional modular protocol stack. In an alternative embodiment, the
25 flexible architecture according to the present invention may be applied to reconfiguration of the RF subsystem 66.

A system is 'adaptive' if it has the ability to reconfigure itself. The baseband according to certain embodiments of the present invention is adaptive.

30 There follows one example of one embodiment of a software architecture according to the present invention, such as for a reconfigurable baseband (R-BB) sub-system is based on Object-Oriented methodology.

Each module 101,102 of the baseband transceiver chain 80 is reconfigurable by instantiation of an appropriate class and/or re-initialisation of module(s) with new attributes (parameters). It is assumed that the software (i.e. class) of each module (e.g. modulator, FEC decoder, etc.) is available (downloaded and stored in the baseband 5 (software) library 70), error free, and compatible. The software download module (SDM) 68 of the SDR terminal is responsible for ensuring that the correct software is downloaded. This is done by setting appropriate parameters that are associated with required software.

Parameterisation of baseband modules is an integral part of the reconfiguration 10 management, in order to set up accurate download requests.

In order to reconfigure the baseband transparently, i.e., without affecting the incumbent services, the baseband architecture must support dynamic creation and binding of new/modified modules. In other words, the baseband architecture must support change in the functionality, behaviour, and the interface of baseband modules on demand without 15 affecting the current operation.

As a result, instantiation of downloaded classes must be administered through dynamic binding, whereby the required functionality of a given class is only made available at run-time, whilst the structure of the downloaded class is known a priori. The software download module 68 of the SDR terminal is responsible for ensuring that the 20 correct software is downloaded. Parameterisation of baseband module classes allows different objects to be derived from the same downloaded class depending upon the agreed new reconfiguration.

Leaf classes are the software blocks which can be used in each of the modules 101, 102. They are downloaded by the software download module 68 and stored in the library 25 70. The software download mechanism needs to ensure that the properties of the leaf classes are compliant with the terminal (type, hardware, user profile, etc.), the standard and the configuration map. Once the leaf class is downloaded and available, the RM 20 will use its valid parameter list to instantiate the required object, and create the required modules 101, 102.

30

Referring again to Fig. 2, and according to an aspect of the present invention, each of the transmit 80a and receive 80b baseband chains comprise an active chain and a

shadow chain. Overall, therefore, baseband transceiver chain 80 comprises an active transceiver chain and a shadow transceiver chain.

The active baseband transceiver chain is the currently operating baseband chain, and comprises an active baseband transmit chain 42 and an active baseband receive chain
5 44.

Each active chain consists of several modules 42a to 42n; 44a to 44n, and each module may contain a Baseband Processing Cell (BPC), according to a further aspect of the present invention, described in further detail below. These modules correspond to the modules 101,102 of Fig.5.

10 The shadow baseband transceiver chain is the post-reconfiguration baseband transceiver chain. That is, it is a baseband transceiver chain which is prepared according to a requested configuration, and will become the active chain once the reconfiguration is activated. It comprises shadow baseband transmit chain 52 and shadow baseband receive chain 54. Rather that duplicating software blocks such as BPCs which are to remain
15 unchanged during reconfiguration, the shadow baseband transceiver chain contains references (pointers) to blocks (BPCs) that are to remain unchanged from the active chain, along with one or more new or modified blocks (BPCs). ,

The shadow transceiver chain 52, 54 is arranged to comply with a reconfiguration strategy which has been agreed by all of the system components involved in the.
20 reconfiguration. The shadow chain 52, 54 does not interfere with the active chain 42,44. The constituent processes are kept mutually independent.

In an embodiment of the present invention, each baseband transceiver chain 42, 44, 52, 54 is made up of several software blocks such as 42-1 to 42-n and 44-1 to 44-n. Each of these blocks may, according to an aspect of the invention, be a baseband
25 processing cell (BPC). The BPC is a fundamental building block of any (active or shadow) transmit-receive baseband chain.

A BPC class is a generic data processing entity. The process class gives the functionality of a BPC object. Several instantiations of the BPC class yields the constituent modules 42a-42n; 44a-44n of the transceiver baseband chain.

30 According to an aspect of the invention, each BPC object (modules 42a-42n; 44a-44n) includes set of Input Ports, I, a set of Output Ports, O, Control Methods, C and a

virtual process function which implements an instance of the downloaded class. This will be further discussed with reference to Figure 6.

Baseband software library 70 contains active and shadow baseband configuration (maps. These correspond to the active and shadow transceiver chains respectively. A 5 configuration map is a list of baseband modules (e.g. baseband processing cells - BPCs) with process definition of each, including type, functionality, algorithmic identity etc., together with interface definitions and interconnections of each BPC. Each baseband configuration map represents the overall definition of the respective baseband, and is a piece of software in itself, which is downloaded when a new standard is to be 10 implemented. In addition, the library 70 will also store all the baseband module classes (the software blocks used, e.g. BPCs) that are currently in use and those that were used previously.

In order to maintain terminal integrity, the library 70 must store either one or both of the following: (a) a 'read-only' default configuration map together with all the 15 associated module classes and parameter lists, which allows the baseband to confidently reconfigure to a known, working standard, which is compliant with the user profile; and (b) a complete copy of the previously working baseband configuration. This latter should include the configuration map, associated baseband classes, parameter lists, operating standard, host network registrations etc. Such a store would allow the terminal to return 20 back to a fully working baseband configuration, without requiring a power off reset.

Reconfiguration switch 78 implements the ON/OFF switch signal 75 from the RM 20, in order to switch the shadow chain 52, 54 ON and the active chain 42, 44 OFF. The shadow chain thereby becomes a new active chain, and the former active chain becomes a shadow chain, ready to be over-written with a new shadow configuration.
25

Several wireless standards may be studied to derive the baseband processing cells, BPCs or classes. These standards may include GSM, EDGE, UTRA-TDD, UTRA-FDD, HIPERLAN/2 DECT, and BLUETOOTH.

According to certain embodiments of the present invention, the formulation of 30 baseband module classes is based on an object-oriented approach. Such formulation aims to identify classes which, when instantiated with appropriate attributes, will help yield the

required baseband functionality in conformance with the configuration map of the standard under operation.

In deriving such BPCs or classes, the following procedure may be adopted. A hierarchical inheritance tree is assumed. The base class of this tree is the Process Class 5 and it provides a virtual method for processing data. It resides at the top of the inheritance tree. Underneath the Process class are classes such as the Parity Check Class, FEC Encoder Class, Modulation Class and so on. The levels of abstraction (i.e., the levels of sub-classes underneath the Process Class) may be developed to suit the desired application. It should be noted that these node classes are abstract, while the leaf classes 10 (i.e., those at the end of the inheritance tree) are the ones that are downloaded as BPC objects or classes.

Examples of leaf classes (BPC objects) include CRC check class, Phase Modulation class, and Convolutional Encoder class. Each leaf class has an associated list of attributes (parameters), which are used to instantiate different objects.

15   Table 1 illustrates some examples of first level of parameters for typical baseband modules, that is, typical baseband leaf classes. These parameters may be extended into organised lists, which can be accessed by the use of an appropriate key.

Baseband Module	Constituent Leaf Class	Attributes
Parity check	CRC check Class	Crc Polynomial, c(x) Crc Size, p
FEC Encoder	Convolutional Encoder Class	Constraint length, k Galois field polynomial, G(p)
Modulation	Phase Modulation Class	Bits per Symbol, n Angle, θ

Table 1

20

Leaf classes (BPC objects) are very specific in their algorithmic detail, yet generic enough to derive unique run-time behaviour and performance through appropriate parameter settings at instantiation.

Object oriented techniques may be used to derive leaf classes through algorithmic 25 abstraction of different baseband modules. This allows different software objects

(modules 42a-42n; 44a-44n) to be instantiated on demand by using the same downloaded leaf class. In other words, different versions of the same algorithm can be generated by instantiating the leaf class with appropriate parameters. This enables very efficient storage in software library and avoids repeated downloads of similar software blocks.

5   The full extent of valid parameters for a given leaf class will depend on the degree of abstraction, and the range of standards which it can be used under, and computer simulations are preferably employed to derive them.

In order to ensure that the correct leaf class is downloaded by the software download module 68, appropriate parameters must be set. These may form the header of
10 the software that is to be downloaded. In addition to algorithmic parameters, each leaf class has an associated property list. This is very important since the SDR terminal's software download mechanism will need to ensure that the software is compatible in its form, fits the agreed configuration map (standard), and provides the desired functionality.

Examples of typical parameters and their associated sub-parameters for baseband
15 software are,

      1. Implementation details

      a. Fixed or Floating point

      b. Hardware features such as processor type, speed, memory requirements

      c. Language, such as C, JAVA, VHDL

20   2. Performance details

      a. Power consumption for a given hardware

      b. Benchmark results for qualitative appraisal

      3. Standard

      a. Identity of the wireless standard that the software is compatible with

25   b. Parameters of the configuration map

      4. Vendor

Software manufacturer identity for compliance check with the terminal manufacturer

      5. Version

30   Version of software, Parity check sum.

These parameters are used by the run-time control signalling handler 72 to ensure that the downloaded leaf class is compatible with the SDR terminal in terms of implementation, performance, standard, vendor authentication and reliability, and version. This check is preferably performed before a software module is stored into the software library 70. It is assumed that the overall terminal management entity EA 64 will be responsible for setting up these parameters, so that the downloaded baseband code is in the correct form, fits with the configuration map and yields the desired performance.

The above-mentioned parameter types may be extended according to the needs of particular wireless communication standards.

Figure 6 illustrates an example of a software block, being a BPC object BPC_A. The algorithmic functionality of BPC_A is defined by the process function 110, which in turn can be configured by its attributes, p 112. Note that p 112 is a list of attributes of the leaf class used to instantiate the process function 110.

As shown in Figure 6, the object BPC_A, contains input ports, I(A) 120 and output ports, O(A) 130. Each port has a unique name, type, and data rate. Synchronous and asynchronous inputs and outputs are provided.

The input ports I(A) 120 comprise a synchronous input port 122 which receives input data 124, and an asynchronous input port 126 which receives runtime status request 128 and runtime signalling 129 inputs. The output ports O(A) 130 comprise a synchronous output port 132 which provides output data 134 and auxiliary outputs 135, and an asynchronous output port 136 which provides a runtime status acknowledge output 138.

The BPC class BPC_A also includes a set of control methods C(A) 140, and they define the state of the BPC object. The process function 110 is in communication with the input and output ports, and the control methods. The RM 20 regulates the state of each BPC object by issuing appropriate signals 201 to the BPC object.

Valid states of each BPC object are illustrated in Figure 7, and include: RUN 305, where the BPC object is processing data as intended; SUSPEND 306, where data is not processed by the BPC object but the object is in a suspended state and is awaiting signal from RM 20 to resume operation; RESUME 304, where data is not output from the BPC object, but the object is awaits signals 201 from the RM 20 to find the correct timing and synchronisation for it to start operating the newly formed BPC object; RESET 307, where

the process function within the BPC object is undergoing re-instantiation wherein the BPC object is not destroyed, but instead changes are made to the process function of the object; INITIALISE 303, where a new BPC object is created; and KILL, where the referred BPC object is to be destroyed and so the BPC object will delete the process function within it.

5   Note that data could be output from the BPC object while in suspend state, provided the functionality of the process function within it supports such an action.

In short, each BPC object performs some digital signal processing, and the type of processing is defined by the particular instantiation of the process class. In turn, a process class can be viewed as a generic data processing class whose algorithmic detail is 10 provided by the leaf classes.

Figure 7 shows the two levels of reconfiguration 301, 302, which can be attained for a given BPC object. The first level 301 relates to the creation and destruction of BPC objects, while the second level 302 relates to the operation and reconfiguration of an individual BPC object.

15   There will now be described certain examples of methods of state transitions of BPC objects, that is, how a BPC object is created and then reconfigured. The two different types 301, 302 of reconfigurations are considered for a BPC object. To aid understanding of the processes involved, pseudo-C++ code is given for each step.

In the following example, it is assumed that the BPC object is a Convolutional 20 Encoder.

At step 303, 'INITIALISE', the RM 20 creates a new BPC object, of identity _id, by instantiating the appropriate BPC class. Pseudo-C++ code for this is:

```
BPC_id = new BPC (Input port addresses, Output port addresses).
```

25

The Convolutional Encoder leaf class code is then read from the software library 70. The RM 20 will create a virtual pointer in its memory and reserve some memory for the new object, which will be instantiated from the class.

The RM 20 will then create a new process function for the new BPC object by 30 issuing the INITIALISE signal. Pseudo-C++ code for this is:

```
BPC_id. INITIALISE (new Convolutional Encoder(k1, G1(p))),
```

where k1 is the constraint length of the code, and G1(p) is the Galois field polynomial. These are the attributes of the Convolutional Encoder Class. Note that, 'INITIALISE' 303 is a member method of the BPC class. Once the BPC object has been

5 initialised, it is then placed by the RM 20 in its appropriate place, as defined by the configuration map.

The RM 20 will then put this new BPC object in RESUME state 304, which implies that the BPC object is almost ready to operate. The RM 20 will determine the correct timing and synchronisation in order to start running the newly created BPC object.

10 Pseudo-C++ code for this is,

BPC_id.RESUME().

Once the RM 20 has established an appropriate time and synchronisation, it will

15 run the new BPC object and place it in the RUN state 305 by issuing appropriate signals 201. Pseudo-C++ code for this is,

BPC_id.RUN().

20   Supposing that this BPC object is to be reconfigured by changing the process function within it. The behaviour or functionality of the convolutional encoder is to be changed without destroying the BPC object. It will remain a convolutional encoder. Such reconfigurations are typical of the type termed 'Partial Reconfiguration' 302.

For such instances, firstly, the RM 20 will make a pointer reference of the BPC

25 object in the shadow chain 52, 54, i.e., *BPC_id. It will then SUSPEND the *BPC_id by placing it in the SUSPEND state 306. Pseudo-C++ code for this action is,

*BPC_id.SUSPEND().

30   Next, the RM 20 will place the BPC object in the RESET state 307, and will perform a reset operation on the BPC object by instantiating a new process function. The new process function implies that there is no change in the input/output ports of the object

*BPC_id, but simply a change in the resident data processing entity, process 110. Pseudo-C++ code for this action is:

        *BPC_id.RESET(new Convolutional Encoder(k2, G2(p))),

5   where k2, G2(p) are the new attributes (parameters) of the process.

Finally, once the BPC object has been RESET, the RM 20 will then put it in RESUME state 304 as discussed above. The RM 20 will next issue a switch ON signal 75 to the reconfiguration switch 78 which will make the shadow configuration active, that is,

10 the active and shadow chains will swap over. Once that is done, the RM 20 will put the reconfigured BPC object in RUN state 305 as discussed above.

Now consider that a new type of FEC encoder (Convolutional Encoder) is required and that the incumbent BPC object is to be replaced by a new BPC object with new input/output ports 120, 130 and a new process 110 within it. Such reconfigurations are

15 typical of the type termed 'Total Reconfiguration' 301.

For such instances, the RM 20 will firstly SUSPEND the BPC object in the shadow chain, placing it in state 306 (similarly to the corresponding step in partial reconfiguration discussed above). The RM 20 will then remove the shadow BPC object by issuing a KILL signal. The KILL signal will move the BPC object into the KILL state

20 308 and will only destroy the process function within it. Once that is successfully completed, the RM 20 will delete the BPC object completely by issuing appropriate signals 201. Pseudo-C++ code for this action is:

        BPC_id.KILL(), and then, delete BPC_id.

25

The RM 20 will replace the killed BPC object with a new one in the shadow chain, which it created in the background by following the steps 'INITIALISE' to 'RUN', as described above.

30   In order to administer reconfiguration, a significant amount of negotiation and management protocols should be maintained. The type of protocols and signalling needed to reconfigure the baseband in a transparent manner are discussed in the following list of

steps, which explain how baseband reconfiguration is initiated, and managed at the baseband in an architecture according to an embodiment of the present invention. Many of the following steps apply to reconfiguration according to the present invention, independently of the possible use of object-oriented techniques, and will apply similarly if
5 software blocks other than BPCs are used.

1. The RM 20 accepts a reconfiguration request from the terminal's equipment authority EA 64, e.g., a Terminal Management Module (TMM)[3]. The request includes information such as,
10   - which BPC objects to reconfigure;
        - how to reconfigure them;
        - configuration map changes;
        - run-time control signal changes/modifications.

15   2. The RM 20 then negotiates the reconfiguration request with the terminal's management entity, that is, equipment authority EA 64. This includes details such as
        - complexity of reconfiguration;
        - processing and memory requirements; and
        - time duration for reconfiguration.
20

3. The RM 20 will perform a RF capability check by referring to the RF property list. It should be noted that the RM is the overall authority of the physical layer and it will control the RF subsystem 66 through appropriate signalling 32.

25   4. The terminal's equipment authority EA 64 will instruct the RM 20 with a list of BPCs identities to be reconfigured, how to reconfigure them and when to reconfigure them.

5. The RM 20 instructs the terminal's equipment authority EA 64 if new software
30 needs to be downloaded, or whether it intends to use the already present software from its local library store 70.

6. The terminal's equipment authority EA 64 instructs the software download module (SDM) 68 if new software needs to be acquired and then instructs the RM 20 when that is available.

7. The RM 20 reads the necessary software from the baseband software library 70. This could be either the newly downloaded code or at that already present.

8. The RM 20 then creates the shadow transceiver chains 52, 54. The shadow chains contain the new baseband modules and pointer references of the unchanged modules, which are intended to remain from the active baseband chain.

9. The RM 20 then validates the shadow chains such that they comply with the agreed configuration map in terms of interfaces between neighbouring BPC objects and their input/output ports.

10. Once the RM 20 has successfully configured the shadow chains, it will then instruct the RF sub-system 66 to re-tune its filters.

11. While the RF sub-system 66 is being reconfigured, the RM 20 sends appropriate SUSPEND and KILL 201 signals to the respective BPCs of the active chains 42, 44. The state transition of BPC objects is then carried out as explained earlier with reference to Fig. 7.

12. The RF sub-system 66 will send an acknowledgement back to the RM 20 after it has successfully reconfigured (e.g. by retuning its filters). In addition, when the active BPCs have been suspended, and successfully killed where required , then the RM 20 will be in a position to switch the shadow chain ON. The reconfiguration switch 78 implements the ON signal 75, to promote the shadow chain to become the active chain; and to move the former active chain to become the shadow chain in preparation for a next reconfiguration.

It has been shown that the timing of this ON signal 75 needs to be authorised by the equipment authority EA 64 in order to maintain network and standard (e.g. GSM, UTRA, DECT) compliance. This is because baseband reconfigurations at the terminal imply corresponding changes at the base station, to which the host network must agree.

5 That can only be possible if the standard under operation permits such flexibility, and has provision for the associated signalling.

The present invention therefore provides a reconfiguration manager 20 for a reconfigurable system; a reconfigurable baseband sub-system R-BB 10; and a

10 reconfigurable software block in the form of a baseband processing cell BPC.

Certain embodiments based on an Object-Oriented rationale have been presented, whereby new baseband modules are created by instantiating appropriate classes. New baseband module classes are preferably downloadable software entities. The management and regulation of the baseband sub-system is the responsibility of the RM 20 of the

15 present invention. The RM is responsible for reconfiguration and run-time behaviour of the whole physical layer, including the baseband sub-system R-BB 10. The RM also interacts with the RF sub-system 66 to check for RF capability, with the EA 64 to negotiate reconfiguration, and with the Software Download Module 68 for acquiring new software.

20

The baseband transceiver chain 80 is made up of constituent BPC objects 101,102; 42,44. Each BPC object provides a unique functionality. The latter is defined by the downloaded classes, which are instantiated within a BPC object. The configuration map defines the baseband chain in terms of functionality of constituent modules and their

25 interfaces and inter-connections. The RM 20 uses this map to create and then connect the different BPC objects. According to an aspect of the present invention, and in order to reconfigure the baseband, the RM creates a shadow baseband chain 52, 54 in compliance with the agreed configuration map for reconfiguration. The switch between the active and shadow is implemented by the RM under the authority of the EA.

30

While the foregoing description provides certain embodiments of the present invention, by way of examples only, numerous modifications and improvements may be

made, within the scope of the present invention. The following improvements are proposed by the inventor.

The interface between baseband 10 and RF 66 sub-systems may be redefined. Different RF architectures may be considered. The extent of digital signal processing (DSP) incorporated within the R-BB will define the range of different BPC objects needed to operate a given standard, representing a trade-off between the number of BPC objects which must be downloaded, versus the cost implications of increased DSP.

The R-BB software architecture may be verified by simulations. Issues of particular relevance are: reconfiguration protocols and signalling; mode switching between different baseband configurations as stated by trial configuration maps; and BPC state management. For the sake of modelling, R-BB 10 will assume the RF sub-system 66 as a software entity, i.e., a class. Instantiations of this class, with appropriate parameters, will allow different RF sub-systems to be created, in software. This exercise is anticipated to help fine-tune the R-BB design by simulating different RF sub-systems. The parameter list for such a class could then be used to derive the RF capability list for the implemented version.

The structure and organisation of baseband software (configuration maps, baseband module classes, and associated parameter lists) within the R-BB library may be modified, as will be apparent to one skilled in the art, without departing from the scope of the present invention as defined in the appended claims.

The present invention also extends to any one or more or the novel features described herein, in any combination.

CLAIMS:

1.   A reconfiguration manager for controlling the reconfiguration of a reconfigurable system, comprising:

5 -   a reconfiguration handler (62) for receiving reconfiguration instructions from a higher authority (64);

-      a run-time control signalling handler (72) responsible for implementing reconfiguration of the reconfigurable system; and

-      a run-time monitoring and status handler (74) for implementing status requests on 10 the reconfigurable system, and for reading acknowledgements and status information from the reconfigurable system.

2.   The reconfiguration manager of claim 1 wherein the reconfiguration handler (62) deals with reconfiguration of the reconfigurable system, including related signalling and 15 negotiations; and is in communication with the higher authority (64).

3.   The reconfiguration manager of claim 1 or claim 2 wherein the run-time control signalling handler (72):

- implements higher layer signalling and control signalling;

20 - deals with run-time signalling, which will be standard specific;

- receives signals from the reconfiguration handler (62) in order to regulate the behaviour of the reconfigurable system while it is being reconfigured;

- provides modifications in the run-time control signalling; and

- is in communication with the higher authority (64), the reconfigurable system (80) and 25 receives signals from the reconfiguration handler (62).

4.   The reconfiguration manager of any of claims 1-3 wherein the run-time monitoring and status handler (74):

- implements run-time status requests on the reconfigurable system (80);

30 - reads in the respective acknowledgements;

- oversees the run-time behaviour of the reconfigurable system;

- collates auxiliary outputs from the reconfigurable system;

- reports to the higher authority (64); and

- is in communication with the higher authority, the reconfiguration handler (62) and the reconfigurable system (80).

5.   The reconfiguration manager of any preceding claim wherein the reconfigurable system comprises a chain of reconfigurable software modules, and the reconfiguration comprises one of:

- total reconfiguration from one standard to another, wherein all software modules are reconfigured; and

10 - partial reconfiguration, which reconfiguration without changing the operating standard, wherein a subset of the modules are reconfigured.

6.   The reconfiguration manager of any preceding claim wherein the reconfigurable system comprises at least one of the baseband subsystem and the RF subsystem of a
15 software defined radio equipment.

7.   The reconfiguration manager of claim 6 wherein the software defined radio equipment comprises a portable terminal incorporating mobile telephony function.

20 8.   The reconfiguration manager of claim 6 wherein the software defined radio equipment comprises a base station for a mobile telephone network.

9.   The reconfiguration manager according to any preceding claim, in combination with a software library (70) for storing software blocks, wherein each of the modules may
25 be loaded with a software block.

10.   The combination of claim 9, wherein the software blocks are received over an air interface by the software defined radio equipment, and stored in the software library (70).

30 11.   The combination according to claim 9 or claim 10 wherein the software library (70) includes a configuration map, the overall module identity, interface, and inter-connection definition list for the standard operating on the equipment.

12.   The combination according to any of claims 9-11wherein a software download module (68) is provided to control the downloaded software blocks.

5 13.   The reconfiguration manager according to claims 6 and 7 wherein the run-time control signalling handler (72) performs signalling to and from the base station; implements the control signalling required by an air interface agreed for use by the operator and the network; and handles all signalling.

10 14.   The reconfiguration manager according to claims 6 and 7 wherein the run-time monitoring and status handler (74) performs internal monitoring of all baseband (80) and RF subsystem (66) operations; checks on each software block (101, 102) of the baseband and/or RF chains; monitors operation of the software blocks; and reports its findings to the higher authority 64.

15

15.   The reconfiguration manager or combination according to any preceding claim wherein the chains of software modules are reconfigured using an object-oriented approach, whereby the functionality of a given software module can be changed by instantiating appropriate classes and/or re-initialisation of module(s) with new attributes 20 (parameters).

16.   The combination according to claim 15 and claim 12 wherein the software download module (68) ensures that the correct software is downloaded by setting appropriate parameters that are associated with required software.

25

17.   The combination according to claim 16 wherein the software download module ensures that the properties of the leaf classes are compliant with the terminal, the standard and the configuration map.

30 18.   The combination according to any of claims 15-17, when dependent on claim 9, wherein, once the leaf class is downloaded and available, the reconfiguration manager

(20) will use a valid parameter list to instantiate each required object, and thereby create the required modules (101, 102).

19. The reconfiguration manager or combination according to any preceding claim
5 wherein each module contains a Baseband Processing Cell (BPC), being a generic data processing entity whose functionality is defined by a process class 110, and the constituent modules (42a-42n; 44a-44n) of the software module chain of the reconfigurable system are provided by several instantiations of the BPC class.

10 20. The reconfiguration manager or combination according to claim 19, wherein each BPC object (modules 42a-42n; 44a-44n) includes set of Input Ports, I, a set of Output Ports, O, Control Methods, C and a virtual process function which implements an instance of the downloaded class.

15 21. The reconfiguration manager or combination according to claim 20, wherein the BPC class also includes a set of control methods (140) which define the state of the BPC object, the process function (110) of the BPC class being in communication with the input and output ports, and the control methods.

20 22. The reconfiguration manager or combination according to claim 21, wherein the reconfiguration manager (20) regulates the state of each BPC object by issuing appropriate signals (201) to the respective BPC object.

23. The reconfiguration manager or combination according to claim 9 or any claim
25 dependent on claim 9, wherein the software library (70) contains active and shadow configuration maps, which correspond to active and shadow software module chains of the reconfigurable system, respectively; each configuration map representing the overall definition of the respective reconfiguration, and being a piece of software in itself, which is downloaded when a new standard is to be implemented.

30

24. The reconfiguration manager or combination according to claim 23, wherein the configuration map comprises:

- a list of software modules with a process definition of each, including type, functionality, algorithmic identity; together with

- interface definitions and interconnections of each software module.

5 25. The reconfiguration manager or combination according to any preceding claim, wherein the reconfigurable system comprises an active software module chain, representing the current configuration, and a shadow software module chain, representing a future reconfiguration, further comprising a reconfiguration switch (78) operable to switch the shadow chain (52, 54) ON and the active chain (42, 44) OFF, the shadow chain 10 thereby becoming a new active chain, and the former active chain becoming a shadow chain, ready to be replaced with a new shadow configuration.

26. A method of performing reconfiguration of a reconfigurable system comprising a plurality of software modules, using a reconfiguration manager according to any preceding 15 claim, wherein the method comprises the steps of:

(a)- the reconfiguration manager (20) accepts a reconfiguration request from the higher authority (64);

(b)- the reconfiguration manager (20) negotiates the reconfiguration request with the higher authority;

20 (c)- the reconfiguration manager (20) performs a capability check by referring to a property list to ensure that the reconfigurable system is capable of operating according to the negotiated reconfiguration;

(d)- the higher authority 64 instructs the reconfiguration manager (20) to reconfigure a list of software modules;

25 (e)- the reconfiguration manager (20) creates a reconfigured version of the reconfigurable system by instructing the reconfiguration of the software modules; and

(f)- the reconfigured system becomes .the active system, and the former configuration of the reconfigurable system becomes unused.

30 27. The method of claim 26 wherein step (e) further comprises the steps of:

(ea)- the reconfiguration manager (20) informs the higher authority whether software needs to be downloaded, or whether it intends to use software already present in a

library (70);

(eb)- the higher authority instructs a software download module (68) to acquire any required software and then instructs the reconfiguration manager (20) when such software is available in the library;

5 (ec)- the reconfiguration manager (20) reads the software from the library.

28.	The method of claim 26 or claim 27 wherein the reconfigurable system comprises an active chain of software modules, representing the currently active configuration, and a shadow chain of software modules, representing a future configuration under construction,

10 or ready for use.

29.	The method of any of claims 26-28 wherein the shadow chain contains new baseband modules and pointer references of the unchanged modules, which are intended to remain from the active chain.

15

30.	The method of any of claims 28-29 wherein step (f) comprises promoting the shadow chain to become the active chain; and to move the former active chain to become the shadow chain in preparation for a next reconfiguration.

20 31.	A method according to claim 28 wherein the reconfiguration manager (20) validates the shadow chains such that they comply with the agreed configuration map in terms of interfaces between neighbouring software modules and their input/output ports.

32.	A method according to any of claims 26-31 wherein step (f) is performed only

25 after authorisation by the higher authority, in order to maintain network and standard compliance.

33.	A method according to any of claims 26-32 wherein the request of step (a) includes one or more of the following pieces of information:

30 - which software modules to reconfigure;

- how to reconfigure them;

- configuration map changes;

- run-time control signal changes/modifications.

34. A method according to any of claims 26-33 wherein the negotiation of step (b) includes negotiation of at least one of the following:

5   - complexity of reconfiguration;

- processing and memory requirements; and

- time duration for reconfiguration.

35. A method according to any of claims 26-34 wherein the instruction of step (d)

10   comprises at least one of the following instructions: a list of software modules to be reconfigured; how to reconfigure them; and when to reconfigure them.

36. The method, reconfiguration manager or combination according to any preceding claim wherein the reconfiguration manager is arranged to perform at least one of the

15   following tasks:

- to define formal procedures and control mechanisms for reconfiguring a reconfigurable system;

- to provide an overall physical layer interface with the higher layers which may themselves be controlled by the higher authority;

20   - to implement protocols for reconfiguration and regulate appropriate signalling to/from the reconfigurable system and also within it;

- to control the baseband and RF sub-systems of an SDR terminal;

- to negotiate reconfiguration requests with the higher authority;

- to carry out a capability list as part of the reconfiguration request negotiations;

25   - to calculate the complexity of a given reconfiguration request;

- to administer reconfiguration of the reconfigurable system following successful completion of negotiations with the higher authority;.

- to interact with a Software Download Module (SDM) of the equipment, for downloading new software;

30   - to maintain a library of software;

- to validate the reconfigured reconfigurable system before switching it on;

- to implement incoming higher layer signalling to the reconfigurable system;

Application No:	GB 0112765.3 3 1	Examiner:	Gareth Griffiths
Claims searched:	All	Date of search:	15 January 2002

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.T): G4A (AFL), H4L (LECCP, LECY, LEP, LERX, LETXX)

Int Cl (Ed.7): G06F 9/44, 9/445, 15/78, H04B 1/40, H04Q 7/32

Other: Online Databases: WPI, EPODOC, JAPIO, INSPEC

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
X, P	WO01/02954 A1 (PHILIPS) whole document	1 at least
X	US5999990 (SHARRIT) whole document	1 at least

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.